TP EJB : Création d'un EJB3 sans état

Table des matières

Propriété du document

Ce document est la propriété de la Société Objis (www.objis.com) , spécialisée dans la formation JAVA/J2EE. Il est fournit aux participants des formations EJB assurées par Objis pour le compte des clients d'Objis.

Prérequis

Les outils et logiciels utilisés se trouvent dans le **répertoire outils** du dossier partagé **FormationEjbObjis**

- JDK 6 (http://java.sun.com)
- Eclipse 3.3 europa Entreprise (http://www.eclipse.org)
- JBOSS 4.2.2 (http://www.jboss.com)

Objectifs

- Créer un EJB3 sans état avec Eclipse

RAPPELS : les raisons de choisir un EJB3 sessions

- Pool d'EJB et meilleur accès concurrents (ex : plusieurs requêtes clients simultanés)
- Possibilité d'accès distant
- Création de services web (basé sur SOAP)
- gestion déclarative de la transaction & de la sécurité
- Services Timers & schédulers
- Intercepteurs (AOP)

Partie 1 : Préparation Eclipse

La spécification EJB 3 simplifie le processus de développement d'un EJB en allégeant le code, de plus les lacunes des EJB Entity sont largement comblées par une nouvelle spécification qui traite spécialement du problème de la persistance : JPA (Java Persistence API).

INFOS : à partir de la version 4.2, JBOSS supporte les EJB3 en natif.

Partie 2 : développement d'un EJB3 session

L'interface

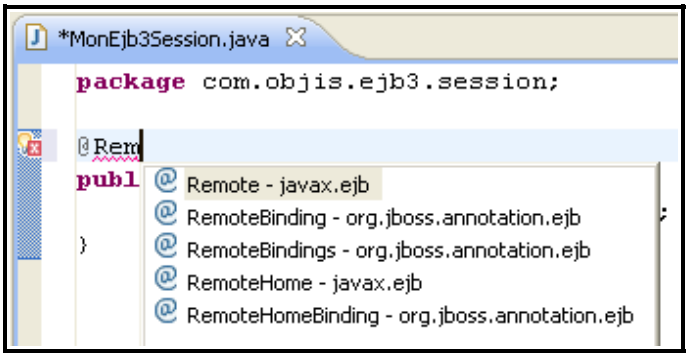
```
package com.objis.ejb3.session;

public interface MonEjb3Session {
	public int addition(int x, int y);
}
```

Le bean

```
package com.objis.ejb3.session;

public class MonEjb3SessionBean implements MonEjb3Session{

	public int addition(int x, int y) {
		return x+y;
	}
}
```

Ajout des annotations

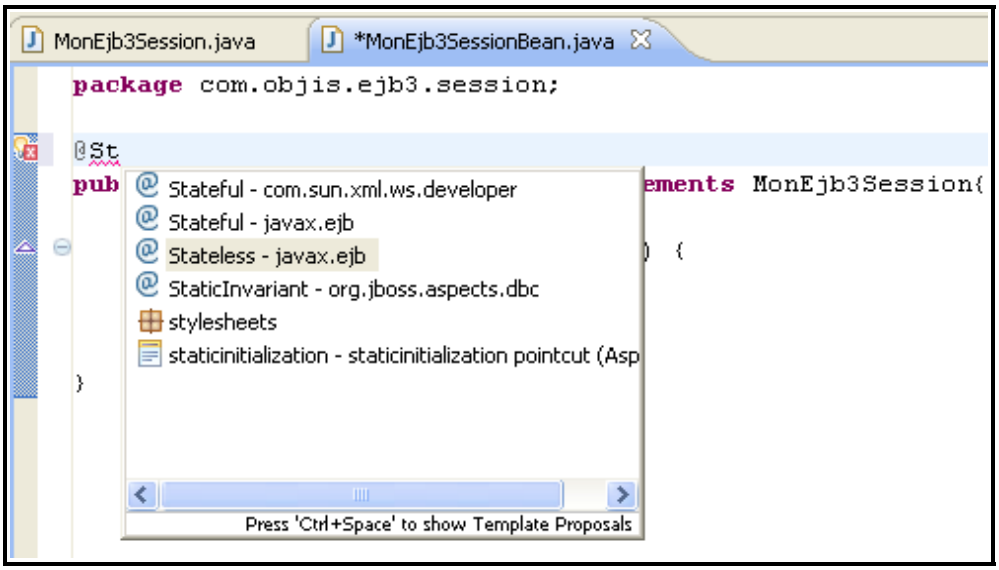
- Ajoutez l' annotation nécessaire **@Remote** pour l'interface. L'éditeur d'Eclipse 3.2 gère complètement la notion d'annotations du JDK 5.0. En particuliers, la complétion (Ctrl+espace) fonctionne pour les annotations et permet d'ajouter simplement la directive 'import' nécessaire



Ce qui donne :

```
package com.objis.ejb3.session;

import javax.ejb.Remote;

@Remote
public interface MonEjb3Session {
      public int addition(int x, int y);
}
```

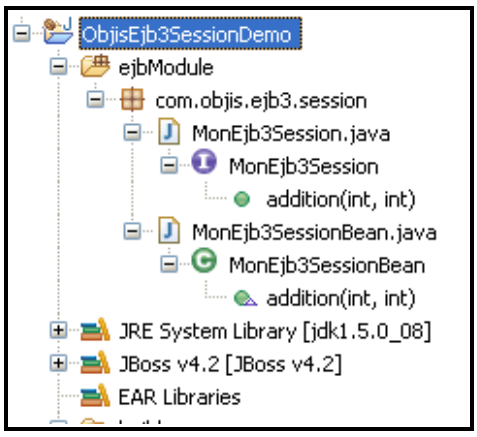
- Ajoutez l'annotation et **@Stateless** pour le bean.



Ce qui donne :

```
package com.objis.ejb3.session;

import javax.ejb.Stateless;

@Stateless
public class MonEjb3SessionBean implements MonEjb3Session{

      public int addition(int x, int y) {
```

```
            return x+y;
        }

}
```

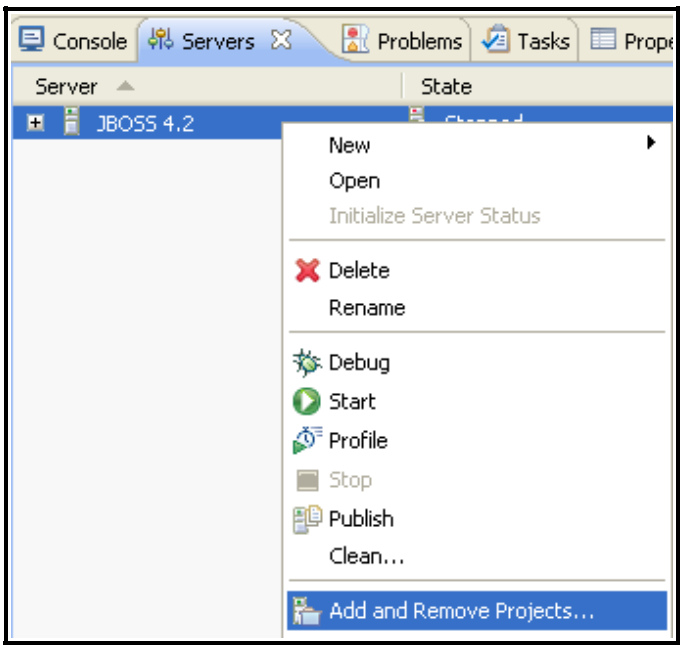


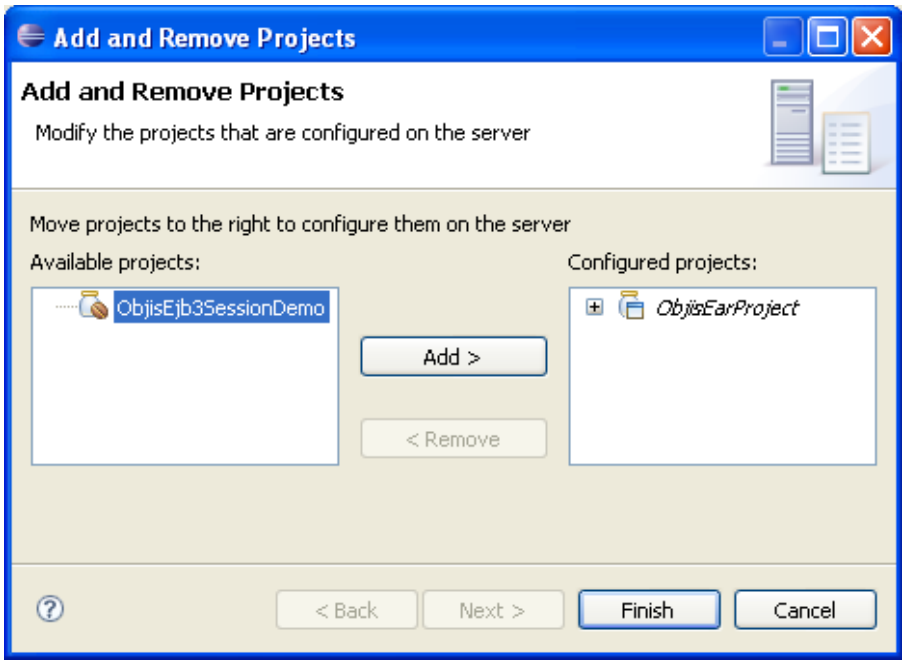
Le dévelopement de l'EJB3 session est terminé.

Partie 3 : test de l'EJB3 session

Le code de notre EJB étant écrit, nous pouvons le tester. Pour ce faire, il faut d'une part déployer le projet EJB dans le serveur JBoss et d'autre part écrire une application cliente.

Déploiement de l'EJB3





Démarrer le serveur JBoss (en mode déboguage de préférence) pour voir si le projet est bien pris en compte.
Si l'EJB est correctement déployé les lignes suivantes doivent apparaître dans la console

```
10:41:47,484 INFO  [JmxKernelAbstraction] creating wrapper delegate for:
org.jboss.ejb3.stateless.StatelessContainer
```

```
10:41:47,500 INFO  [JmxKernelAbstraction] installing MBean:
jboss.j2ee:jar=ObjisEjb3SessionDemo.jar,name=MonEjb3SessionBean,service=EJB3
with dependencies:
10:41:47,734 INFO  [EJBContainer] STARTED EJB:
com.objis.ejb3.session.MonEjb3SessionBean ejbName: MonEjb3SessionBean
10:41:47,812 INFO  [EJB3Deployer] Deployed: file:/D:/outils/jboss-
4.2.2.GA/server/default/deploy/ObjisEjb3SessionDemo.jar
```

Création application cliente

Créez un projet JAVA '**ObjisEjb3SessionDemoClient**'

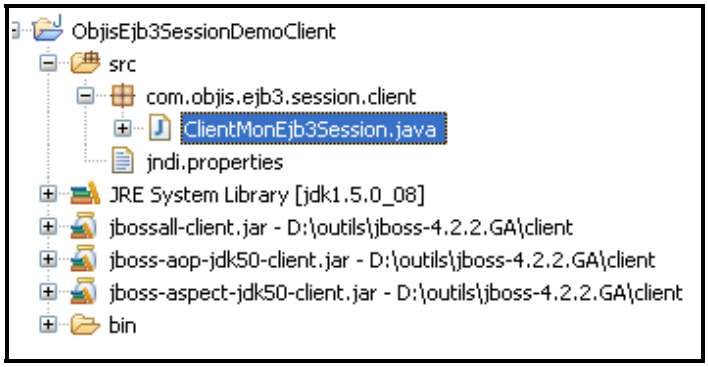
A la racine du projet créer un fichier nommé **jndi.properties** contenant les informations qui permetteront à l'application cliente de se connecter au service de nommage du serveur JBoss :

```
java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
java.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interfaces
java.naming.provider.url=localhost:1099
```

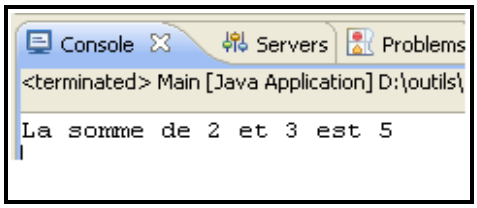
- Créez la classe suivante

```
package com.objis.ejb3.session.client;

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

import com.objis.ejb3.session.MonEjb3Session;

public class ClientMonEjb3Session {

      public static void main(String[] args) {
            try {
                  Context context = new InitialContext();
               MonEjb3Session beanRemote = (MonEjb3Session)
               context.lookup("MonEjb3SessionBean/remote");
               System.out.println("La somme de 2 et 3 est " +
beanRemote.addition(2, 3));
            } catch (NamingException e) {
                  e.printStackTrace();
            }
      }

}
```

Pour l'exécution : sélectionner la classe dans la vue '**Explorateur de projets**' et utiliser le menu '**Exécuter->Exécuter en tant que->Application Java**'



Le lancement de l'application donne :



Meilleures pratiques EJB3 Sessions

1. Bien choisir son type d'EJB Session (Statefull ou Stateless ?).
2. Bien examiner le type d'interface. Si le client de l'EJB est dans la même JVM, utiliser @Local
3. Séparer les préocupations transversales (Log, audit) à travers des intercepteurs
4. Examiner le type de données seront sauvegardées dans un échange conversationnel. Valorisez les petites données
5. Ne pas oublier les méthodes @postCreation, @preDestroy (pour ex : libérer les ressources) ainsi que (pour EJB Stateful) @passivation, @activation

FIN